Supporting Materials

Inherent Vacuity for GR(1) Specifications

Introduction

In this folder you can find:

1. A pdf document with additional detailed that are not part of the paper submission. See file “**vacuityPaperSupport.pdf**”
2. Raw data for the evaluation section of the paper. See folder “**RawData**”.
3. Ways to reproduce the raw data. See folder “**VacuityTest**”.

Raw Data Description

Throughout the files we use the following codes for groups of vacuities. We sometimes refer to these groups by the term “categories”.

“EMOD” stands for specification element vacuities of the environment.

“SMOD” stands for specification element vacuities of the system.

“EVAL” stands for domain values of the environment.

“SVAL” stands for domain values of the system.

Folder “**RawData/Vacuities**” contains running time results for sets of specifications. All runs are performed 10 times for each specification.

For each set of specifications there are 5 files:

Files with suffix “_SAT” contain three columns.

Column A contains the filename.

Column B has “TRUE” for satisfiable specifications, and “FALSE” otherwise.

Column C contains running time of the satisfiability check in milliseconds.

Files with other suffixes, namely “_EMOD”, “_SMOD”, “_EVAL”, and “_SVAL” contain running times for *satisfiable* specifications only for the category of specifications its name indicates.

Column A contains the filename.

Column B contains the time to first vacuity in milliseconds.

Column C contains the time to all vacuities in milliseconds.

Column D contains the number of all vacuities.

Column E (only for “_EMOD” and “_SMOD”) contains the number of trivial vacuities.

Folder “**RawData/Lists**” contain lists of vacuities found in each specification set. In each file that has the specification set name and prefix “_list” there are 5 lines for each file. The first line is the filename.

Each of the other lines begin with “EMOD” “SMOD”, “EVAL”, or “SVAL”. After that there is a number denoting the number of vacuities (-1 if the specification is unsatisfiable), and identifiers of the vacuities. For specification element vacuities the identifier is its line number. For domain value vacuities the identifier has the format “var_name=value”.

Folder “**RawData/Cores**” contain files with data about core computations of confirmed vacuities. We have 10 runs for each core computation. In each file that has the specification set name and prefix “_CORES” there are the following columns:

Column A contains the filename.

Column B contains the category of the vacuity for which we compute the core.

Column C contains the ID of the vacuity (line number or “var_name=value”).

Column D contains the type of the element (“Ini” for initials, “Safe” for safeties, and “Just” for justices).

Column E contains the core size (0 for trivial vacuities).

Column F contains the premise-set size

Column G contains running time of the core computation in milliseconds.

Column H contains line numbers of the core members.

Folder “**RawData/Impact**” contain files with data about synthesis running times before and after the removal of a vacuity.

Columns A-D are the same as in “**RawData/Cores**”

Column E contains synthesis running time of the original specification

in milliseconds.

Column F contains synthesis running time of the specification without the vacuity in milliseconds.

Note that we only have information for folders of realizable specifications because unrealizable specifications cannot be synthesized.

Note also that raw data files contain information about domain value vacuities (for which we added a safety that ensures the domain value is never set, instead of removing a specification element), but those were not reported.

Folder “**RawData/Validity**” contain files with data about validity of vacuities. In each file that has the specification set name and prefix “_VALID” there are the following columns:

Columns A-D are the same as in “**RawData/Cores**”

Columns E-I contain report if the following properties are invariant when the vacuity is removed: realizability, satisfiability, satisfiability of the environment module, satisfiability of the system module, and well-separation.

Producing Raw Data

Folder “**VacuityTest**” contains an executable, sets of specifications in Spectra format, and scripts, that allow the reproduction of raw data files on computers running windows 10, with Java 8 64bit and Microsoft Visual C++ 2015 Redistributable (x64) available here:
https://www.microsoft.com/en-us/download/details.aspx?id=48145

Running times may differ according to the computation power of the computer one uses.

How to Install and use

Copy folder “**VacuityTest**” anywhere on your hard drive. Installation is done. This allows you to:

a. Run script files (files with "bat" extension) that produce the csv files with the raw data we used for our experiments.

b. Directly run executable jar files we provide, on Spectra specifications we provide (the files with the ".spectra" extension)

c. Run executable jar files on any other Spectra specification, with different parameter values.

In order to run java jar files open a windows console, change directory to the one containing the jar file and write

```
java -jar VacuityTestExec.jar PARAMS
```

PARAMS are the required parameters as explained bellow, under “Using the executable directly”.

Sets of specifications

Folders that start with “SYNTECH” contain SYNTECH specification sets described in the corpus. The suffix “_C” on some of them means that they are cleared from non-GR(1) specifications as described in the paper.

Folders “cimattirealizable” and “cimattiunrealizable” contain the AMBA and GENBUF specifications described in the paper.

We also added the folder “LIFT” which contains six files with the lift specification in our running example that had a specification modeling a controller for a lift operating on 3 floors, in file “Lift3.spectra”, and scaled versions of it with up to 8 floors, the last one being “Lift8.spectra”.

Scripts

For convenience there are three scrips that create raw data from the specifications.

The script “ALL10” creates files in folder “Vacuities”.

The script “ListsAndCores10” creates files in folders “Lists” and “Cores”.

The script “ListsAndImpact10” creates files in folders “Lists” and “Impact”.

The script “ListsAndValidation” creates files in folders “Lists” and “Validity”.

Using the executable directly

By making additional folders with other files in spectra format, it is possible to use our executable to create data on other specifications, and choose the number of times we repeat runs on files.

Using command line, get into the folder “**VacuityTest**” and write

```
java –jar VacuityTestExec.jar    FOLDER    TEST_TYPE   [NUM]
```

FOLDER   is the name of the folder containing the set of specifications in spectra format.

TEST_TYPE   is:

ALL_VAC - for “Vacuities” files generation

LIST - for “Lists” files generation

CORES - for “Cores” files generation

IMPACT - for “Impact” files generation

VALID - for “Validity” files generation

The last three options require a list of vacuities file for the specification set. These files have the suffix “_list” and they must be in the “Lists” folder. In order to run them one must first run the executable with the LIST option, or copy one that was already prepared into the “Lists” folder.

For the TEST_TYPE parameters “ALL_VAC”, “CORES”, and “IMPACT” one must also provide the NUM option, which is a natural number controlling the number of runs we have for each instance.

For example, the line

```
java –jar VacuityTestExec.jar    Lift    ALL_VAC  3
```

runs the files in the Lift folder. It creates 5 files under the “Vacuities” folder for each. One file is with suffix “_SAT” for satisfiability and another four for running times and number of vacuities in each category. Each specification is run 3 times.

As another example, the line

```
java –jar VacuityTestExec.jar    cimattirealizable CORES 10
```

requires the file “cimattirealizable_list” in folder “Lists” and runs 10 core computations for each documented vacuity in that file. This results in a file “cimattirealizable_CORES” in folder “Cores” with the data.

In order to produce the file “cimattirealizable_list” required for this run, one should use the command

```
java –jar VacuityTestExec.jar    cimattirealizable LIST
```